Am Luneort 15 a
27572 Bremerhaven
Germany

Operating Manual

Navigating European Marine Observer (NEMO)

Data Decoding

Part Number: 160 0 00 00 004

Version: 100916

Revision No:	IR02
Revision Date:	2011-01-26

THIS PAGE IS INTENTIONALLY LEFT BLANK

RECORD OF REVISIONS

This page lists the manual revisions. When you receive revision pages, replace the old pages with the new and log the information in this table.

Title	NEMO Data Decoding
Part Number	160 0 00 00 004
Version	100916

REVISION NUMBER	ISSUE DATE	CHANGE	INCORPORATION DATE	INCORPORATED BY

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE IS INTENTIONALLY LEFT BLANK

1 DESCRIPTION OF IRIDIUM/GPS TRANSMISSION

1.1 General

The Iridium SBD Modem can send and receive SBD messages. The data sent are called 'Mobile Originated Message' (MOM), data received are called 'Mobile Terminated Message' (MTM).

The Iridium SBD message has a maximum length of 205 bytes for MOM and 135 bytes for MTM. The profile data do not fit into one SBD message. Therefore several SBD messages have to be sent. This is organized in a data structure, consisting of a header and the user data.

1.2 Mobile Originated Message (MOM)

The table below shows the organization of a MOM with header (Byte 01-08) and user data (Byte 09-205).

Byte #	***Data type***	***Parameter***	***Description***
01	unsigned char	crc_status	Status of CRC calculation: 'Y' = ok, 'N' = failure.
02 – 03	unsigned int	checksum	Checksum of MOM data.
04	unsigned char	mtmsn	Successfully received MTM message number.
05	unsigned char	elementary_num	Elementary message number of total messages.
06	unsigned char	total_num	Total number of messages.
07	unsigned char	device_data	Device- and datatype (see table below).
08	unsigned char	profilenumber_lsb	Least significant byte of current profile number .
09 – 205	unsigned char	data	Array of MOM user data bytes.

The device and data type byte has the following legend.

Nibble	***Bits***	***Typ***	***Description***
Upper	08 – 05	Device type	Always 1 for NEMO floats.
Lower	04 – 01	Data type	0 = Test message 1 = Profile data message 3 = Recovery data message 4 = Manual file transmission

1.3 Mobile Terminated Message (MTM).

The table below shows the organization of a MTM with the header (Byte 01-04) and the user data (Byte 05-135). The MTMSN helps the operator to identify whether the MTM command is transmitted successfully or still queued. On success the MTMSN sent is in the next MOM header.

Byte #	***Data type***	***Parameter***	***Description***
01	unsigned char	crc_status	Status of CRC calculation: 'Y' = ok, 'N' = failure.
02 – 03	unsigned int	checksum	Checksum of MTM data.
04	unsigned char	mtmsn	MTMSN for actual MTM message.
05 – 135	unsigned char	data	Array of MTM user data bytes.

1.4 MTM commands

To send commands to the NEMO the operator has to use the MTM format. The commands are in ASCII format in the data array and terminated by carriage return <CR> and line feed <LF>.

At the moment only three commands are available.

- profileabortion — limits are 0 - 999
- recoverydelay — limits are 1 – 10080, unit: minutes
- parkingdepth — limits are 9 – (profile pressure), unit: dbar/10
- profiledepth — limits are 10 – 2000, unit: dbar/10

NOTE: THE ASCII COMMANDS ARE CASE SENSITIVE.

For example, a complete MTM in hex-format with the MTM command 'profileabortion=1<CR><LF>'.

59 34 12 02 70 72 6F 66 69 6C 65 61 62 6F 72 74 69 6F 6E 3D 31 0D 0A

The checksum of the MTM is 1234_{hex}. The MTMSN is 02. The data consists of ASCII values (profileabortion=1<CR><LF>).

1.5 How to use the remote configuration

The remote configuration only works when the status messages are transmitted by the instrument (at start procedure and in tracking mode). You need the International Mobile Equipment Identification (IMEI) of the transmitter and an email account.

Sending a new configuration to a NEMO Float:

- Create a new text file with *.sbd extension.
- Write the commands to execute in this file and save.
- Create a new email to data@sbd.iridium.com .

Subject: the IMEI of the NEMO float, i.e. 300034012209890

- Attach the text file you created to the email.
- Transmit the email.

The buoy will receive the message during the transmission of the next status message.

THIS PAGE IS INTENTIONALLY LEFT BLANK

2 DECODING/CONVERSION OF RAW DATA

2.1 General

The data structure of the NEMO Float is in the following format:

Profile header (fixed length)
Data set 1 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 2 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 3 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 4 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)
Data set 5 (variable length) Data record 1 (fixed length) Data record 2 (fixed length) " " " " Data record 65535 (fixed length)

The profile header is located at the beginning of the data structure and contains the basic information of the instrument and the size of the different data sets. The profile header has a fixed length (see chapter 2.4 for more details).

The five data sets have a variable length and contain a variable number of data records. The quantity of data records inside a data set is stored in the profile header. There can be a maximum of 65535 data records in one data set. The data records itself contain the raw data with a fixed length, one measurement sample.

For data processing the following data types are used:

unsigned char	=> 1 Byte
unsigned int	=> 2 Bytes
signed int	=> 2 Bytes
unsigned long	=> 4 Bytes
signed long	=> 4 Bytes
double	=> 4 bytes

NOTE: ALL DATA STORAGE IS IN LITTLE ENDIAN FORMAT.

2.2 Data structure of profile data

For the profile data only two of the five data sets are used. 2

The data structure of the mission profile data is in the following format:

- Profile header
- Data set 1: unused
- Data set 2: parking data (CTD)
- Data set 3: parking position data (RAFOS)
- Data set 4: profile data (CTD, O2)
- Data set 5: extended Surface data (GPS)

2.3 Data structure of Iridium test message

The Iridium test message is also stored within the general data structure.

For the Iridium test message only two of the five data sets are used.

- Data set 1: configuration and housekeeping parameters of instrument
- Data set 2: compact surface data (GPS)
- Data set 3: unused
- Data set 4: unused
- Data set 5: unused

2.4 Format of profile header

Byte #	***Data type***	***Parameter***	***Description***
01 – 02	unsigned int	profilenumber	Profile number
03 – 04	unsigned int	serial_number	Serial number of device.
05	unsigned char	upcast_status	Status of profile upcast. 0 = UPCAST_BUSY 1 = UPCAST_DONE 2 = UPCAST_TIMEOUT 3 = UPCAST_ICE_DETECTION
06	unsigned char	not_used	Not used.
07 – 08	unsigned int	older_profiles	Number of older profiles not sent.
09	unsigned char	motor_errors	Number of motor errors during profile.
10	unsigned char	sbe41_errors	Number of SBE41 CTD errors during profile.
11	unsigned char	o2optode_errors	Number of O_2 - Optode errors during profile. Zero if no sensor installed.
12	unsigned char	rafos_errors	Number of RAFOS errors during profile. Zero if no rafos module installed.
13	unsigned char	cpu_battery	CPU battery voltage in profile depth.
14	unsigned char	pmp_battery	Pump battery voltage in profile depth.
15	unsigned char	motor_current	Motor current in profile depth.

Byte #	*Data type*	*Parameter*	*Description*
16	unsigned char	motor_current_mean	Mean of motor current during upcast.
17 – 18	unsigned int	internal_pressure_surface	Internal tube pressure [mbar] at surface.
19 – 20	unsigned int	internal_pressure_depth	Internal tube pressure [mbar] in profile depth.
21 – 22	signed int	not_used	Always zero.
23 – 24	unsigned int	not_used	Always zero.
25 – 26	unsigned int	pressure_offset	Offset pressure of the CTD.
27 – 28	unsigned int	surface_pressure	CTD surface pressure before descent.
29 – 30	unsigned int	parking_pressure_median	Pressure at end of parking (median).
31 – 32	unsigned int	depth_pressure	Pressure before ascent.
33 – 34	unsigned int	depth_pressure_max	Max pressure during mission.
35 – 36	unsigned int	profile_recovery	Profile number when mission will be aborted and recovery mode will be entered.
37 – 46	unsigned int	data_record_quantity	Quantity of the five different data records
47 – 48	unsigned int	piston_counts_surface	Piston position at surface.
49 – 50	unsigned int	piston_counts_parking	Piston position in parking depth.
51 – 52	unsigned int	piston_counts_calc	Piston position in parking depth calculated.
53 – 54	unsigned int	piston_counts_depth	Piston position in profile depth.
55 – 56	unsigned int	piston_counts_eop	Piston position at end of profile.
57 – 60	unsigned long	descent_start_time	Time when profile starts [seconds].
61 – 64	unsigned long	parking_start_time	Time when parking starts [seconds].
65 – 68	unsigned long	upcast_start_time	Time when upcast starts [seconds].
69 – 72	unsigned long	ascent_start_time	Time when ascent starts [seconds].

just before descent

Byte #	Data type	Parameter	Description
73 – 76	unsigned long	ascent_end_time	Time when ascent ends [seconds].
77 – 80	unsigned long	surface_start_time	Time when surface starts [seconds].
81 – 90	unsigned int	data_record_quantity_copy	Copy of quantity of the five different data records (redundant)

2.5 Format of data set 'profile data'

For the ascent profile the data does not come in sequence (see chapter 2.1).

Byte #	Data type	Parameter	Description
01	unsigned char	rtc_hour	RTC hour of data record.
02	unsigned char	rtc_minute	RTC minute of data record.
03	unsigned char	rtc_second	RTC second of data record.
04	unsigned char	not_used	Not used.
05 – 06	signed int	sbe41data_pressure	CTD pressure.
07 – 08	unsigned int	sbe41data_ temperature	CTD temperature.
09 – 10	unsigned int	sbe41data_ salinity	CTD salinity.
11	unsigned char	o2_format	O_2 - Optode format*
12	unsigned char	not_used	Not used
13 - 14	unsigned int	o2_oxygen	O_2 - Optode concentration. †
15 - 16	unsigned int	o2_saturation dependent	O_2 - Optode saturation. †
17 - 18	unsigned int	o2_temperature ⇒ für diagn.	O_2 - Optode temperature. ✓
19 - 20	unsigned int	o2_dphase dependent	O_2 - Optode rawdata †
21 - 22	unsigned int	o2_bphase bei 3830 Optode	O_2 - Optode rawdata ✓
23 - 24	unsigned int	o2_rphase	O_2 - Optode rawdata ?
25 - 28	unsigned long	o2_bamp	O_2 - Optode rawdata †
29 - 30	unsigned int	o2_bpot	O_2 - Optode rawdata †
31 - 34	unsigned long	o2_ramp	O_2 - Optode rawdata †
35 - 38	signed long	o2_rawtem	O_2 - Optode rawdata †

tc phase bei 4330(F) Optode

concentration = depend

Byte #	*Data type*	*Parameter*	*Description*

* depends on settings of the O_2- Optode: Format 0: bytes 13-18 , Format 1: all bytes.

2.6 Format of data set 'parking data'

Byte #	*Data type*	*Parameter*	*Description*
01 – 02	signed int	sbe41data_pressure	CTD pressure.
03 – 04	unsigned int	sbe41data_ temperature	CTD temperature.
05 – 06	unsigned int	sbe41data_ salinity	CTD salinity.

2.7 Format of data set 'position data'

The data set 'position data' consists of RAFOS and CTD data.

Byte #	*Data type*	*Parameter*	*Description*
01	unsigned char	status	Status of RAFOS dataset. 2 = OK 4 = OVERLAP ERROR 8 = ON @ END ERROR 16 = COMMUNICATION ERROR
02	unsigned char	rtc_year	RTC year at start of data record.
03	unsigned char	rtc_month	RTC month at start of data record.
04	unsigned char	rtc_day	RTC day at start of data record.
05	unsigned char	rtc_hour	RTC hour at start of data record.
06	unsigned char	rtc_minute	RTC minute at start of data record.
07	unsigned char	rtc_second	RTC second at start of data record.
08 – 13	unsigned char	rafos_amplitude	Six values of amplitudes from the RAFOS module.
14	unsigned char	not_used	Not used.
15 – 26	unsigned int	rafos_rank	Six values of corresponding ranks

Byte #	Data type	Parameter	Description
			from the RAFOS module.
27 – 28	signed int	sbe41data_pressure	CTD pressure.
29 – 30	unsigned int	sbe41data_ temperature	CTD temperature.
31 – 32	unsigned int	sbe41data_ salinity	CTD salinity.

2.8 Format of data set 'compact surface data'

The data set 'compact surface data' consists of GPS date/time an position data.

Byte #	Data type	Parameter	Description
01	unsigned char	gps_year	GPS year of fix.
02	unsigned char	gps_month	GPS month of fix.
03	unsigned char	gps_day	GPS day of fix.
04	unsigned char	gps_hour	GPS hour of fix.
05	unsigned char	gps_minute	GPS minute of fix.
06	unsigned char	not_used	Not used
07 – 08	signed int	latitude	Latitude of fix (N=positive, S=negative).
09 – 10	signed int	longitude	Longitude of fix (O=positive, W=negative).

2.9 Format of data set 'extended surface data'

The data set 'surface data' consists of current Real-Time Clock (RTC) and GPS date/time as well as position data.

Byte #	Data type	Parameter	Description
01	unsigned char	rtc_year	RTC year of fix.
02	unsigned char	rtc_month	RTC month of fix.
03	unsigned char	rtc_day	RTC day of fix.
04	unsigned char	rtc_hour	RTC hour of fix.
05	unsigned char	rtc_minute	RTC minute of fix.
06	unsigned char	rtc_second	RTC second of fix.

07	unsigned char	gps_year	GPS year of fix.
08	unsigned char	gps_month	GPS month of fix.
09	unsigned char	gps_day	GPS day of fix.
10	unsigned char	gps_hour	GPS hour of fix.
11	unsigned char	gps_minute	GPS minute of fix.
12	unsigned char	gps_second	GPS second of fix.
13 – 16	signed long	latitude	Latitude of fix (N=positive, S=negative).
17 – 20	signed long	longitude	Longitude of fix (O=positive, W=negative).

2.10 Format of data set 'configuration and housekeeping data'

The configuration data consists of all setup parameters of the instrument.

Byte #	***Data type***	***Parameter***	***Description***
01 – 02	unsigned int	serial_number	Serial number of instrument.
03	unsigned char	software_year	Software version year
04	unsigned char	software_month	Software version month
05	unsigned char	software_day	Software version day
06	unsigned char	rtc_year	Current RTC year
07	unsigned char	rtc_month	Current RTC month
08	unsigned char	rtc_day	Current RTC day
09	unsigned char	rtc_hour	Current RTC hour
10	unsigned char	rtc_minute	Current RTC minute
11 – 12	unsigned int	piston_counts_max	Piston position at inflated oil bladder (for maximum depth)
13 – 14	unsigned int	piston_counts_min	Piston position at deflated oil bladder (for surface).
15	unsigned char	cpu_battery	CPU battery voltage
16	unsigned char	pmp_battery	Pump battery voltage
17	unsigned char	internal_pressure	Internal tube pressure
18	unsigned char	not_used	Not used.

Byte #	*Data type*	*Parameter*	*Description*
19 – 20	unsigned int	profile_pressure	Profile pressure in dbar/10
21 – 22	unsigned int	parking_pressure	Parking pressure in dbar/10
23 – 24	unsigned int	transmission_time_max	Transmission time in minutes
25 – 26	unsigned int	parking_sample_interval	Interval for parking measurements in minutes.
27 – 28	unsigned int	mission_cycle_time	Mission cycle time in minutes.
29	unsigned char	descent_mode	Use last piston position until this profile number.
30	unsigned char	not_used	Not used.
31 – 32	unsigned int	profile_abortion	Enter recovery mode after this profile number.
33 – 34	unsigned int	descent_speed	Descent speed during downcast in mm/s.
35 – 36	signed int	ice_temperature	Ice temperature in [degree Celsius/1e3].
37	unsigned char	recovery_transmission	Maximum transmission time for recovery message in [minutes].
38	unsigned char	not_used	Not used.
39 – 40	unsigned int	recovery_delay	Delay time between recovery messages in [minutes]

2.11 Data conversion

Battery voltage:

The battery voltage is measured with 12-bit resolution.

The converted 8-bit value has one decimal place (e.g. value = 143 => 14.3 V).

Hydraulic current:

The hydraulic current is measured with 12-bit resolution.

The converted 8-bit value has two decimal places (e.g. value = 80 => 0.80 A).

The maximum current value is 2.55 A).

Internal tube pressure:

The tube pressure is measured with 12-bit resolution.

For 'profile header' the converted 16-bit value with no decimal place (e.g. value = 863 => 863mbar) is used.

The 'configuration and housekeeping data' format uses the converted 8-bit value with one decimal place (e.g. value = 105 => 10.5 psi).

Piston position:

The piston position value is measured with 12-bit resolution.

The converted 16-bit value has no decimal place (e.g. value = 3245 => 3245 counts).

There maybe an offset for min/max piston position value. This depends on the mechanical configuration of the specific instrument.

CTD data:

All CTD data (temperature, salinity and pressure) for park and profile are stored with two bytes. The CTD pressure data has no offset correction. The CTD pressure offset value is stored within Profile Header and has to be added for further calculations. The CTD pressure offset value is measured every time the Float is at surface.

CTD-temperature: 5 digits, three decimal places (1milli-degree resolution).

CTD-salinity: 5 digits, three decimal places (1milli-psu resolution).

CTD-pressure: 5 digits, one decimal place (10cm resolution).

	Hex-value	Dec-value	Converted	Unit
Temperature:	3EA6	16038	16.04	°C
Temperature*:	F58B	2677	-2.677	°C
Salinity:	8FDD	36829	36.829	PSU
Pressure:	1D4C	7500	750.0	dbar
Pressure**:	FFF1	65521	-1.5	dbar

*Note regarding negative temperatures (T °C < 0).

**Note regarding negative pressure. Pressure values are signed values.

Positive temperature range is 0 to 62.535°C (0 to F447 hex).

Negative temperature range is -0.001 to -3.000°C (FFFF to F448 hex).

If(hex value) >= F447, then calculate FFFF – (hex value) + 1= Y_DEG.

Convert Y from hex to decimal, divide by 1000, and multiply by (-1), for degree C.

Positive pressure range is 0 to 3276.7 (0 to 7FFF hex) dbar. Negative pressure range is -0,1 to -3276.8 dbar (FFFF to 8000). If (hex value) > 7FFF, then calculate FFFF – (hex value) +1 = Y_PRES.

Convert Y from hex to decimal, divide by 10, and multiply by (-1), for pressure in dbar.

O2 data:

Depending on the configuration of the optode (see user manual) the data can include only processed data (concentration, saturation, temperature) or additional raw data. The accuracy and an example is shown below.

O2 concentration: 5 digits, two decimal places, offset of 10.

O2 saturation: 5 digits, two decimal places, offset of 10.

O2 temperature: 5 digits, two decimal places, offset of 10.

O2 Dphase: 5 digits, two decimal places.

O2 Bphase: 5 digits, two decimal places
O2 Rphase: 5 digits, two decimal places
O2 Bamp: 9 digits, one decimal place
O2 Bpot: 5 digits, two decimal places
O2 Ramp: 9 digits, one decimal place
O2 RawTem: 9 digits, two decimal places

	Hex-value	Dec-value	Converted	Unit
Concentration:	658F	25999	249.99	uMol/l
Saturation:	2A48	10824	98.24	%
Temperature:	2C7	711	-2.89	°C
Dphase, Bphase, Rphase, Bpot:	7F0A	32522	325.22	
Bamp, Ramp:	0016DA4C	1497676	14976.76	
RawTem	FFA6CF26	5845210	58452.10	

GPS data:

GPS time and date information is stored with one byte each.

GPS geographic position information (latitude/longitude) is stored with 2 (compact surface data) or 4 (extended surface data) bytes each.

Table 1: Example for value conversions of GPS data.

	Hex-value	***Dec-value***	***Converted***	***Unit***
GPS year:	06	6	6	
GPS month:	09	9	9	
GPS day:	18	24	24	
GPS hour:	7	7	7	
GPS minute:	20	32	32	
GPS second:	1E	30	30	
GPS latitude:	14E6	5350	53.5	degree
GPS longitude:	359	857	8.57	degree
GPS latitude*:	1FE3E461	535028833	53.5028833	degree
GPS longitude*:	051C2F2E	85733166	8.5733166	degree

*Note regarding extended surface data.

2.12 Checksum

All calculated checksums are using the 16-bit CRC-CCITT algorithm with following parameters:

Polynomial function:	$x^{16} + x^{12} + x^5 + 1$
Polynomial:	0x1021
Initial remainder:	0xFFFF
Final XOR value:	0x0000

No reflection of data

No reflection of remainder